ISSN 0254–2757

QUÆSTIONES INFORMATICÆ

Volume 6 • Number 4 April 1989

The official journal of the Computer Society of South Africa and of the South African Institute of Computer Scientists

Die amptelike vaktydskrif van die Rekenaarvereniging van Suid-Afrika en van die Suid-Afrikaanse Instituut van Rekenaarwetenskaplikes

QUÆSTIONES INFORMATICÆ

The official journal of the Computer Society of South Africa and of the South African Institute of Computer Scientists

Die amptelike vaktydskrif van die Rekenaarvereniging van Suid-Afrika en van die Suid-Afrikaanse Instituut van Rekenaarwetenskaplikes

Editor

Professor D G Kourie
Department of Computer Science
University of the Pretoria
Hatfield
0083

Production
Mr Q H Gee
Department of Computer Science
University of the Witwatersrand
Johannesburg
Wits
2050

Subscriptions

The annual subscription is

	SA	US	UK
Individuals	R20	$ 7	£ 5
Institutions	R30	$14	£10

to be sent to:
Computer Society of South Africa
Box 1714 Halfway House 1685

Quæstiones Informaticæ is prepared by the Computer Science Department of the University of the Witwatersrand and printed by Printed Matter, for the Computer Society of South Africa and the South African Institute of Computer Scientists.

Editorial

by

Derrick Kourie

It is my privilege to have been requested by the SAICS executive to take over the post of editor of QI from Professor Judy Bishop. I think it is in order to thank her on behalf of the readership for the fine job she has done in boosting the quality of the journal during her brief but effective term. It is also appropriate to thank the production editor, Quintin Gee, for his substantial role in producing the journal. I am grateful that he is still in the post, and for all the support and work that he continues to do.

My job as editor is directed towards the overall goal of serving the South African academic community in the various computer-related disciplines in particular, and the computer industry in general. A number of objectives which support this goal include

- ensuring that high quality papers are published, thereby providing a display window for computer-related research in South Africa
- boosting local and international circulation of the journal both within the academic community and in the computer industry at large, thereby promoting a fruitful interchange of ideas
- attempting to do this in a cost-effective fashion so that the limited financial resources of SAICS and the CSSA may be released (perhaps even modestly augmented) to promote their various other service-orientated activities.

A number of measures are planned which are intended to meet these objectives. I shall mention some of them below, while others will become manifest with the passage of time.

After much debate it has been decided to change the name of this journal from Quæstiones Informaticæ to The South African Computer Journal/Die Suid-Afrikaanse Rekenaartydskrif. It will be abbreviated to SACJ in English and SART in Afrikaans. Arguments against this name change include the conciseness and uniformity of reference in both official languages provided by QI, and a certain kind of catchiness to the name. Those in favour of the name change regard the new proposal as being more descriptive for ordinary mortals (i.e. non-Latin scholars), less pretentious, and therefore more inviting for a wider audience. The fact that the new title identifies the journal as South African is also regarded as important. Many readers would, I surmise, be fairly neutral about the name and adopt a philosophical "a rose by any name" position. Perhaps the divide is between those who opt for a high level of abstraction and information hiding, and those who feel that a measure of refinement is necessary.

Regarding the quality of papers, I shall continually strive to ensure that papers submitted are reviewed by at least two relevant and competent specialists. It is appropriate here to thank all those who have so enthusiastically reviewed papers to date. This is a time-consuming, altruistic, backroom task, with very little explicit reward. To ensure that the burden is spread more equitably, I would like to appeal to readers to suggest additional names of people who could be approached for reviewing. Names of overseas contacts would be particularly useful.

I should also like to invite as much reader-participation in the journal as possible. There are several levels at which this may be done. The most obvious is by way of letters to the editor. Many people out there have strong ideas about a variety of subjects. In the absence of a decent national network facility (perhaps someday!), please feel free to use SACJ as your soapbox.

However, it is also evident that many people read many books for a variety of purposes. Why not share these insights by submitting book reviews to the journal, particularly with respect to books which could be prescribed for courses? If there are any book publishers or distributors out there who perchance may read this editorial, perhaps you should make inspection copies to lecturers contingent on a review being provided to SACJ!

I would also encourage researchers to continue providing a steady stream of research papers to the journal. Clearly, SACJ is in competition with other international journals for your research results. However, this is not a head-on competition. While it would be sheer hubris to pretend that SACJ is precisely equivalent to one of the more prestigious overseas publications, there are considerations which argue in favour of submitting certain kinds of research to SACJ. First, SACJ will be dedicated to providing a quick turnaround in reviewing and publication. Hence, it is an ideal forum for presenting and testing interim research results, and even for quickly assuring your stamp on potentially important ideas which you hope to flesh out later. Secondly, SACJ is the obvious forum to use for locally relevant research. Finally, and quite candidly, the competition for publication in SACJ is obviously not as intense as in a more prestigious international journal. However, I need to be most

explicit on the implications of this latter point.

SACJ should not be seen as a soft option in the sense that quality will be sacrificed. By this I mean that on some arbitrary scale of quality measurement, if CACM contains papers above say the 95% percentile, then SACJ should fall into about a 60% percentile category. Put differently, there is clearly a gap to be filled that lies somewhere between poor, inferior drivel and outstanding research contributions – a gap which SACJ will seek to fill. Papers will therefore be rigorously reviewed, and every effort will be made to ensure that the journal is worthy of international recognition – even if such recognition does not come about immediately. This is not the impossible task that some might consider it to be. There are several South African scientific journals that already enjoy a measure of international recognition (the South African Statistical Journal – to name but one). Furthermore, it is my perception that many of our academics who travel overseas discover – perhaps slightly to their amazement – that they are well able to hold their own with academics at peer institutions. This suggests that there is probably sufficient brain power, research ability and research activity in the country to ensure that the goal of international recognition is attained.

As for the cost-effective functioning of SACJ, two points need to be made. First, SACJ will be available for a limited amount of advertising at R1000 per page and R500 per half-page. The computer industry and book publishers might wish to avail themselves of this offer, as might universities and employment agencies. Enquiries in this regard should be directed to Quintin Gee. Secondly, a modest charge per page (indicated elsewhere in this edition) will be levied on accepted research papers. This has become standard practice for most journals, the rationale being that the SACJ is one of the journals which counts for state subsidy purposes. However, the editor will have the right to waive such charges in deserving cases, as for example in the case of an author from industry whose company is unwilling to provide the financial support.

Ultimately then, SACJ will critically depend on your support. It will become what you, the reader, researcher and reviewer, make it. In a sense the South African Computer Journal will expose you, the South African Computer Academic, to the outside world without a single Latin phrase to hide behind.

Reflections on the Nature and Future of Computer Science in Southern Africa

B H Venter
Department of Applied Computer Science, University of Fort Hare, Private Bag X1314, Alice, Republic of Ciskei

Received September 1988, Accepted October 1988

1. Introduction

Computer Science is a relatively recent scientific discipline. It has seen rapid growth and even more rapid change. Not surprisingly there is constant controversy among computer scientists about the nature of the discipline, and how best to pursue it.

This article aims to stir up the controversy. The views expressed in it are personal, and the article is an adaptation of the author's inaugural address at Fort Hare University.

2. What's in a Name?

Before going on to more serious matters, I wish to reflect on the name "Computer Science". The name has been the subject of one of the most heated and least productive controversies, and Fort Hare has not escaped.

To get some idea of how low the level of debating can become, consider the following statement which one hears far too often

> "If a discipline has to attach the word science to its name, it's probably not a science. For example:
>
> Chemistry, Physics, Geology, etc.
>
> as against
>
> Social Science, Political Science, Computer Science."

When confronted with such a statement many Computer Scientists react by coming up with names like Computerology, Informatics, and so on. I believe this is misguided because:

- Changing the name of the discipline is not going to change its nature.
- It is terribly narrow-minded to contend or accept that the classic disciplines of the natural sciences are the only truly "scientific" disciplines.

Of course, for the most part, people wish to change the name of the discipline because they believe that it does not accurately reflect the nature of the discipline. Hence one often hears names like Computing Science, Programming Science, Information Science, Information Processing, Software Engineering, and so on.

It also happens that a university department changes or extends the term "Computer Science" in order to indicate that it is somehow different or superior to the "plain vanilla" computer science departments. For example, the Fort Hare department is currently known as the department of "Applied Computer Science." I suspect that it acquired this name in order to place some distance between the curriculum of the department and the very Numerical Mathematics oriented curricula of departments at other universities in the early seventies. The distinction has since faded considerably.

I do not believe that playing around with the name of the discipline or an individual department serves any useful purpose. The term "Computer Science" has become entrenched and is not likely to be changed. Virtually every computer science department in Southern African is called "Department of Computer Science" and I believe Fort Hare should follow suit.

3. Is Computer Science a Natural Science?

A less heated but far more serious controversy concerns the fundamental nature of Computer Science. Is it a natural science, or an engineering discipline, or the underlying discipline of a profession distinct from Natural Scientist or Professional Engineer? Put in another way: does a computer science department belong in a natural science faculty, an engineering faculty, a commerce faculty, or a faculty of its own?

Put in yet another way, should a South African professional computer scientist be registered in terms of the act covering natural scientists or the act covering professional engineers, or should there be a special act, or should there not be any act at all?

In the United States computer science departments have ended up in natural science faculties as well as in engineering faculties, depending on whether the "parent" department had been Mathematics or Electrical Engineering. Apparently, these departments flourish equally well in either faculty.

In South Africa, most departments find themselves in natural science faculties. Correspondingly, the

South African Council for Natural Scientists (SACNAS) attempted to list "Computer Scientist" as one of the professions covered by its act.

On the other hand, SACNAS did not, at that stage, recognize the South African Institute for Computer Scientists (SAICS), and hardly consulted SAICS about the listing. Officially this happened because SAICS did not have enough qualified members when it sought recognition. However, one cannot help but suspect that SACNAS did not really wish to find a way around this difficulty. Perhaps, some of the senior natural scientists involved in the issue were not quite convinced that Computer Science is a natural science.

The intended listing of "Computer Scientist" as a profession reserved for registered natural scientists was met by an outcry from the myriad of non-graduates who make a living from programming computers. Few of these people would qualify for registration as natural scientists and they quite rightly fear that listing "Computer Scientist" may eventually lead to their exclusion. Of course, the employers of these people are even more horrified at the prospect of having to compete for the services of a very small pool of academically qualified people.

Needless to say, the masses prevailed and "Computer Scientist" was removed from the list of professions reserved for registered natural scientists.

My own opinions about these issues are:

- Computer Science may well not be a true "natural science" because it is not chiefly concerned with natural phenomena, However, if that is the case, then one should exclude the Mathematical sciences as well.
- Computer Scientists ought not to register with SACNAS, and SACNAS probably ought not to exist in the first place. The computer software industry, for one, is not yet ready for a limited entry profession, and public safety will be better served by making software producers more accountable for the quality and consequences of their products.
- I personally feel most comfortable in a natural science faculty, and students from such faculties tend to fare better with Computer Science than students from other faculties. Moreover, in Southern Africa, the structured curricula of engineering and commerce faculties are simply too crowded to adequately accommodate the topics a professional computer scientist should know.

  I feel further more that the difference between Computer Science and Physics is no greater than the difference between Physics and Zoology or between Zoology and Chemistry. And, on the whole, I feel that Computer Science is more similar to the natural sciences than to any other grouping of sciences.

4. Do we need Computer Scientists in Southern Africa?

Computer science departments largely base their curricula on a model drawn up in the United States, and succeed in producing graduates that are internationally acceptable as computer scientists. This is no empty claim, for quite a few of our graduates proceed overseas to further their studies, and they generally do quite well there. Unfortunately, some never return.

One of the reasons for this mini brain drain may well be a lack of jobs for computer scientists (A "Computer Scientist", of course, is someone with at least an honours degree). Certainly, many of the industries that employ computer scientists in developed countries have no local equivalents. Instead, the bulk of computer related employment in Southern Africa is offered by the data processing departments of companies that have little interest in computers other than using them to implement business information systems.

In the past, the management of some of these data processing departments have been sharply critical of the curricula of computer science departments. In effect, their criticisms amounted to the following : local industry has very little use for computer scientists, and instead needs commerce graduates with a sound computer background, augmented by non-graduate programmers trained with specific programming skills.

Such complaints and representations have led to the creation of departments of "Business Data Processing" in the commerce faculties of universities. Furthermore, Technikons and private training companies have rushed in to provide training to the legions of non-graduate programmers demanded by data processing departments.

Such developments appear to eliminate data processing departments as a reason for producing computer scientists. What's left of local industry is mainly the growing number of companies that develop complex systems incorporating computers. These companies certainly have more use for computer scientists. However, they are typically dominated by engineers and their managers claim to need "software engineers" rather than computer scientists.

Not surprisingly, engineering faculties have rushed to include programming courses in their curricula, and they are starting to claim that software engineers should be produced by engineering faculties rather than by science faculties.

One may thus reasonably ask whether it makes sense to go on producing computer scientists when those that do not emigrate end up with employers that supposedly have little use for computer scientists. Would it not make more sense to distribute the handful of computer science lecturers

among engineering and commerce faculties, and forget about having computer science departments?

My view is that the following reasons are more than adequate to justify the existence of computer science departments in their current form:

- A BSc with Computer Science as a major subject is a useful education even for students that do not go on to become active computer scientists.
- Computer Science, in its present "scientific" form, is a sensible minor subject for any bachelor's degree, since it thoroughly demystifies the computer. It is also an effective way of developing problem-solving skills, methodical thinking, and the ability to abstract.
- Constructing computer software is an extremely demanding task. Much of what is now done by non-graduates should rather be done by computer scientists. Employers are starting to realize this.
- The curricula of engineering and commerce faculties are too crowded to allow their students to acquire all the skills and knowledge needed to become effective software developers.
- Eventually, the availability of professional computer scientists will lead to industry undertaking more projects that cannot possibly be carried out by non-computer scientists.

5. The State of Computer Science Research in Southern Africa

Coming back to the computer scientists that further their studies overseas and then stay there. I guess that at least a few of them become computer science researchers, and stay overseas because they despair of their chances to do research in Southern Africa. Certainly, it is true that university lecturers are expected to do research, and many *do* research, but it is also true that lecturers get very little time for research and very little recognition for what they manage to do.

And while it is true that most computer science departments are understaffed, this does not mean that a prospective researcher can easily find a suitable job upon graduating. Right now, Fort Hare would probably be unable to hire such a graduate.

To the best of my knowledge, no computer science research is currently being conducted outside of universities. It may well be that I am simply not informed about it, but it is also certain that such research is neither published in the research journals nor publicized elsewhere.

Until a year ago, the CSIR maintained a computer science department as part of the, now defunct, National Research Institute for Mathematical Sciences. This department, while under-equipped and understaffed by international standards, had a staff including more than 10 professional computer scientists (including myself), most of whom were pursuing higher degrees.

This made the department larger than any university department apart form UNISA. Correspondingly, almost a third of all the papers presented at the most recent South African Computer Symposium originated from this department.

Unfortunately, the publishable research conducted by this department was severely impeded by management attitudes and priorities, as well as by the lack of senior staff of international stature. The department was thus unable to establish computer science research on a firm and respectable footing in Southern Africa.

Nevertheless, while it existed, the department had the potential of doing this. Unfortunately, the demise of the CSIR as a body carrying out research with its own staff has squandered this potential. The newly appointed chairman of the governing council of the CSIR recently motivated this as follows (my translation):

> "Initially this (research carried out by the CSIR) was the best approach, because South Africa needed a strong scientific infrastructure.
>
> Now, however, we have reached the stage where the universities can truly stand on their own, and the basic sciences are firmly rooted.
>
> The CSIR thus need not duplicate what the universities are doing, or are able to do, and can rather use its enormous pool of talent in the service of industry."

Following this change of strategy by the CSIR, many of its computer scientists (including myself) left its service. Those that did not, or could not, were transferred to the CSIR computer centre (data processing department) and told to earn money somehow. I doubt whether much publishable computer science research will ever again originate from the CSIR.

Presumably, the current state of computer science research at the universities, had little influence on the making of such momentous decisions, In fact, computer science issues probably do not figure in many decisions at all. The low status of computer scientists is perhaps best summarized in the introduction to the proceedings of the last South African Computer Symposium, written by Prof. Pieter Kritzinger of UCT:

> "I know of no computer scientists in South Africa who is in a position where (s)he can affect funding priorities. As far as I know we have no representation on any of the committees of the Foundation for Research and Development, and I know of no-one in our Afrikaans speaking fraternity who is a member of the Akademie vir Wetenskap en Kuns. It will take time and conscious effort to establish our presence. The same is true of course for our universities. Again, with one exception, I know

of no dean of a science faculty, vice-principal or principal who is a computer scientist."

As I have already mentioned, computer science lecturers have little time available for research. This problem is greatly compounded by the very nature of computer science research: all computer science research is ultimately concerned with the construction of software. More often than not, a research project requires a great deal of software to be specially constructed, and constructing software is extremely time consuming. The result is that years of work may culminate in a single research paper. In fact, in reading research in my fields of interest, I seldom encounter the same author twice, except when the same paper is published over and over again.

Coupled to the low stature of computer scientists, this trickle of papers per individual researcher makes it almost impossible, if not completely impossible, for a computer scientist to obtain significant funding from the Foundation for Research and Development, in competition with established researchers in other fields, who in some cases manage to produce as many as ten papers per year.

The result is insufficient equipment, insufficient library allocations, insufficient opportunities to attend research conferences, insufficient post-graduate students, insufficient research-oriented computer science lecturers, and so on. Which in turn perpetuates the insufficient research. I regret to conclude that computer science research is in a sorry state in Southern Africa. Moreover, the signs are that things are getting worse rather than better.

6. Do we Actually need to do Computer Science Research in Southern Africa?

One may justifiably ask if it really matters whether or not computer science research is carried out in Southern Africa. Lecturers need to do research in order to be able to train research students. But if these students cannot find jobs as researchers, apart from overseas or as lecturers, why go to all this trouble? Why not simply educate students up to the honours level and be done with it?

Well, one obvious answer is that one is going to have trouble finding and keeping university lecturers if this is the attitude taken, and this may already be happening. But in addition to that, I feel that a thorough training in basic research is an excellent background for doing development and implementation.

That is, we may not need computer science researchers outside of the universities, but we do need people that were trained as computer science researchers. Besides, the availability of enough trained researchers may in itself lead to the creation of a need for researchers.

7. So what do we do about it?

One is always tempted to pronounce "the central government must do something about this, or there will be dire consequences to the national well-being." And certainly, in the case of South Africa, one may reasonably ask whether a country that can support an Antarctic research team, and that is contemplating a space program, will not be better off sacrificing one of these in favour of comprehensively supporting computer science research.

However, from the bleak picture I have painted thus far, I think it is clear that public sector and private sector policy makers are not likely to come to the aid of Computer Science in Southern Africa.

It is thus up to policy makers in the universities to provide the necessary support for Computer Science. It is largely thanks to the efforts of these people that Computer Science was established in Southern Africa in the first place, and it will largely be up to them to keep it viable in the future.

Of course, it is ultimately up to the computer scientists themselves to do so well with *is* available, that the discipline will ultimately generate its own growth, funding and respectability.

Abstracts: MSc/PhD Conference held at Dikhololo in 1988

The 3rd Conference for MSc/PhD computer science students, held at Dikhololo in 1988, was attended by representatives of most South African Computer Science departments. The conference, which was sponsored by the RCP Group and organised by the Computer Science Department of the University of South Africa, was worth attending, and included some very interesting research activities. Abstracts of the papers presented at the conference are produced below.

The Specification of an Operating System

H C Ackerman
University of Stellenbosch

Abstract

Computer hardware has improved dramatically over the past decade. Current technology offers processing potential on a chip which was traditionally associated with mainframe computers. Software science development has unfortunately not kept up with engineering progress, and the development of systems today is handicapped by software reliability rather than hardware.

The correctness of large software systems depends on three main factors: correct decomposition of the system into modules; individual correctness of the modules; and correctness of the interfaces between component modules. This paper is concerned with the latter: formal specification of module interfaces.

Ideally a system should be decomposed into component modules which are implemented in parallel by several programmers. Experience shows that errors are caused by the integration of modules, indicating that the formalism used for the specification of decomposed models is inadequate. It should be possible to check the correctness of a specification mechanically in order to rule out human error in the proof of specification correctness.

Algebraic specification offers a formal method of specifying systems top-down, and several specification languages for algebraic specification are in use today. We examine the feasibility of a comprehensive algebraic specification of a non-trivial operating system.

A Semantic Interpreter for LOTOS

D Behr
University of Pretoria

Abstract

System specifications for communication and distributed systems using FDTs like LOTOS have been receiving a lot of international attention. These systems tend to be concurrent, asynchronous and non-deterministic. Specifying and verifying such systems using traditional techniques is difficult. As part of a research project at UP, an environment for the specification of communication and distributed systems is being developed. The part of the project discussed here deals with the development of a semantic interpreter for LOTOS specifications. The semantic interpreter generates traces for a specification. These traces provide a means of verifying the LOTOS specification. International attention tends to prove a trace as either acceptable, or not acceptable to a specification. This approach however tends to provide another means of characterizing a system, by generating a sufficiently big subset of traces and/or subtraces. These provide an alternate means of verifying aspects of a specification.

The Feasibility of Applying Expert System Technology to the System Performance Analysis and Tuning of Operating Systems

D A Bryant
University of Stellenbosch

Abstract

It is proposed that an expert system that can be applied to analyse the performance of IBM mainframe computer systems running under the OS/VS2 MVS operating system should be developed. The expert system will utilize performance data logged by the standard MVS software monitors such as SMF, RMF, GTF and the CICS monitoring facility. The expert system will analyze this data and report on the inefficient usage and performance degradation of the system. The expert system will identify exception events and provide the most likely cause of the event, together with the actions recommended to remedy the situation. The knowledge embedded in such a system will represent the skills of the most expensive and scarce programmers in the computer industry.

In order to evaluate the feasibility of this approach, three prototypical expert systems have been developed. Each assesses the ASM or Auxiliary Storage Management component of the OS/VS2 MVS operating system. The placement and size allocation of the data sets on the DASD devices within the system configuration are evaluated, possible system bottlenecks are highlighted, and recommendation to improve system performance are made.

The first expert system developed makes use of Goldworks, an expert system development tool, or expert system shell based on Golden Common Lisp software. The second uses the environment provided by Arity Prolog, an extended version of the Prolog programming language. The final expert system has been developed using a procedural language, Modula-2, to mimic the actions of the AI languages.

This paper evaluates the relative applicability and suitability of each of the three approaches to the particular problem area under consideration.

Speeding Up Ray-Tracing

R F Breedt
CSIR

Abstract

Ray-tracing is an elegant approach to realistic image generation. Shadows, reflections, transparency and other visual effects can all be handled in an integrated way. However, being a brute force method, ray-tracing is a very slow process. Speed-ups can be achieved by exploiting the inherent parallelism of the process and by using approximate data structures.

This paper introduces the concept of ray-tracing and then discusses various software and hardware solutions for reducing the time complexity of ray-tracing.

Image Processing Applications on the Transputer

N Cooke
Rhodes University

Abstract

This paper describes an investigation into the application of Image Processing techniques on a transputer network hosted by a PC-AT. The research concentrates on identifying an appropriate distribution of tasks between the network and the 80286 processor for Image Processing applications. The major considerations in the investigation are benchwork of standard Image Processing techniques

on the two processing systems and the comparison of their architectural features which support Image Processing. To complement the investigation, a system has been developed for displaying images from either of the processors with the aid of an Enhanced Graphics Adapter. The results of the research are used to draw conclusions about the viability of this hardware configuration as an Image Processing workstation.

EXPROG: 'n Outomatiese Programmeringstelsel

J P du Plessis
Universiteit Oranje Vrystaat

Abstract

EXPROG is 'n outomatiese programmeringstelsel wat Pascal programme genereer. EXPROG maak gebruik van die kennis-gebaseerde benadering tot probleemoplossing. Gevolglik besit EXPROG 'n verskeidenheid soorte (tipes) kennis wat deur middel van verskeie voorstellingsmetodes, bv. reëls, rame en EXPROG definisies, voorgestel word. Die gebruiker beskryf, in Engels, die spesifikasie van die verwagte program aan EXPROG. EXPROG redeneer dan met behulp van sy bestaande kennis totdat 'n algoritme gekonstrueer is wat voldoen aan die spesifikasie soos verskaf. Daarna word die algoritme gekodeer in Pascal om sodoende die verwagte program daar te stel.

A Modula-2 Implementation for the Transputer

W J Hayes
CSIR

Abstract

This paper reviews message passing techniques for communication between processes. An implementation of Modula-2 for the Transputer is then given. Finally, added features to Modula-2, providing concurrent programming and message passing facilities, are discussed.

Analytic Modelling Approaches for DASD Subsystems

H Joubert

Modelling of Shared Resources in Computer Systems: The Multiserver Station with Concurrent Customer Classes

S Crosby
University of Stellenbosch

Abstract

With the sponsorship of the SA Post Office and Semantyx (Pty Ltd.) a group of students at the University of Stellenbosch is undertaking research into the modelling of DASD subsystems of large computer systems. Research involves the investigation, application and development of appropriate mathematical techniques to model DASD subsystems to a high degree of accuracy. The group will produce a modelling package with a graphics interface based on existing windowing software, which will allow easy specification and modification of models.

Performance monitoring data from the DASD subsystem to be modelled will be used for automatic

construction of the model, and results of the model will be displayed graphically.

Two papers will be presented, which will give an overview of the project and cover in some detail the existing techniques for, and proposed research into, mathematical modelling of DASD subsystems.

A Survey of Standards for the Exchange of Digital Geographical Information

A Cooper
CSIR

Abstract

There are a number of countries and international organizations who are developing standards for the exchange of digital geographical information. Some of these standards attempt to cater for all types of geographical information while others are aimed at specialized subsets of geographical information, such as cadastral or topographic information. While some standards have been implemented, none of them have been tested completely.

The author has been involved in drafting the South African exchange standard and has held discussions with people involved in the drafting of the standards of the United States, the United Kingdom and the International Hydrographic Organization. This paper will discuss these and other exchange standards.

Parallel Process Placement

C Handler
Rhodes University

Abstract

This paper investigates methods of automatic allocation of processes to available processors in a given network configuration.

The major considerations which have been addressed in deciding upon a particular allocation scheme are:

- the number of processes to be allocated versus the number of processors available in the network;
- the computing power of each processor;
- the work load of each process; and
- the interdependency between processes, e.g. the use of shared variables and the degree of inter-process communication.

The research work involves the implementation and testing of various algorithms for optimal process allocation, as well as the gathering of performance statistics during program execution for use in improving subsequent allocations.

The system has been implemented on a network of loosely-coupled microcomputers using multi-port serial communication links to simulate a transputer network. The concurrent programming language Occam has been implemented replacing the explicit process allocation constructs with an automatic placement algorithm, enabling the source code to be completely separated from the hardware considerations.

A Knowledge-based Design Environment for Information Systems

J Kambanis
University of South Africa

Abstract

This paper describes the functionality and architecture of an Information Systems design environment which is based on a Semantic Data Model. Recent research in Software Engineering has recognised the promising possibilities of using Knowledge Representation techniques from Artificial Intelligence, in

modelling Information Systems. The environment described is based on this approach. It integrates User Interface techniques used in Computer Aided Software Engineering and Knowledge Acquisition Environments to support modelling of Information Systems. Knowledge representation is based on the Taxis language development at the University of Toronto. The architecture of the environment is aimed particularly at maximising its generality and extensibility.

The Synthesis and Processing of Waveforms to Generate Musically Interesting Sounds

A J Kesterton
Rhodes University

Abstract

Audio frequency sound of musical interest can be generated by converting a set of data from the digital to the analog domain. Various standard methods exist to generate the data and alter it to create more musically interesting sounds. These methods can be implemented in hardware or software or both.

This paper gives details of Frequency Modulation and Additive Synthesis algorithms being implemented entirely in software to generate sound data. The implementation of various filters and the direct manipulation of the digital data are also discussed. The downloading of the resultant data, using the MIDI protocol, to a sampler (a type of digital music synthesizer) to perform the conversion to the analogue domain is described.

The software is designed to run both as a stand-alone system and as part of a Music Network system being implemented at Rhodes. The Network uses the XINU real-time operating system and Ethernet hardware on IBM PC look-alikes.

Sellulêre Outomate

L Kotzé
Randse Afrikaanse Universiteit

Abstract

Een- en twee-dimensionele sellulêre outomate word gedefinieer. Verskillende eienskappe van sellulêre outomate word ondersoek; onder andere die volgende:

- Die patrone wat voortgebring word deur die outomate onder verskillende begintoestande.
- Die ooreenkomste tussen twee- en meer-toestand outomate. Die ooreenkomste tussen sellulêre outomate en PASCAL se driehoek.
- Die afledingsvermoë van sellulêre outomate.

Toepassings vir sellulêre outomate in die fisika, chemie en ander velde word bespreek.

An Expert System for Diagnosing, Tuning and Optimising Computer Systems Performance

W H Kotze
University of Stellenbosch

Abstract

The expert system VAXEXP evolved from the need to support the systems and performance management of large and complex VAX systems. There was also a need to provide some training support for junior systems staff.

The initial starting point is the VAX system(s) where a combination of Fortran, VAX DCL and System Services collects the required data and concentrates it into one large ASCII file. This is ported across to a microcomputer supporting Arity

Prolog under MS/DOS 3.1. VAXEXP is planned to consist of three main modules, the first of which, called SYSGEN, will provide information on the usage of the systems parameters, indicate the implications of the value of a particular parameter, and lastly uses the data from the VAX to suggest a better suited value for a parameter. This module is nearly finished and contains some 350 rules.

The second main module, called PERFLEARN, is designed to be an aid in building up expertise that will enable VAXEXP (and the system manager) to turn a qualified statement into a quantified rule, e.g. "the direct I/O is EXCESSIVE" → "If the direct I/O is xxxx THEN it is EXCESSIVE".

The final module, called PERFEVAL, will actually make extensive use of the data from the VAX to identify and diagnose existing and potential problems and where possible will also suggest a remedy. There are some 500 potential rules for this module.

Query Processing in Distributed Database Systems

S M Lamprecht
University of Stellenbosch

Abstract

Query processing in distributed database systems involves the determination of an execution strategy. An execution strategy is a sequence of operations that must be performed on the database in order to obtain the required result to solve the query. For a given query several execution strategies can be ascertained. It is therefore a very important problem in query processing to acquire the optimal execution strategy. An execution strategy is optimal with respect to a selected cost function. This cost function is determined by several different factors. One of the most important factors in the distribution environment is the cost of transmitting data between different computers. Several optimisation techniques exist which optimise data transmission cost. Some of these optimisation techniques will be presented.

Exploiting Redundancy in Knowledge Representations

R Layton
University of the Witwatersrand

Abstract

Most complex representational formations embody both explicit and implicit redundancy. Much effort has been extended at both identifying this redundancy and reducing it to some minimal form.

Isolated cases have proposed that redundance be exploited within specific areas of knowledge such as machine vision and sensor-based diagnostics but no overall model has yet been proposed.

A model is proposed that exploits both implicit and explicit redundancy in order to maximise consistency-checking and to facilitate multiple viewpoints in a single representation. This leads to a conclusion that, whereas a universally accepted representation cannot exist, a hybrid model with controlled redundancy exhibits properties required of such a universal representation.

A Communication Kernel for a Distributed Database System

M D Meumann
University of Stellenbosch

Abstract

The major components of a Distributed Database Management System comprise, amongst others, a network facility enabling access to remote sites and a concurrency control mechanism to handle and

synchronize distributed transactions. These two features are the major components of the communication kernel of a DDBMS currently under development at the University of Stellenbosch using the XENIX operating system. This talk will describe the networking features being included in the XENIX operating system using ArcNet communication cards and the concurrency control being implemented based on a conservative timestamp algorithm.

An Algorithm Development System

C C Pienaar
University of Stellenbosch

Abstract

A short informal introduction is given to the weakest precondition methodology for developing an algorithm and a proof of its correctness, with reference to other popular programme verification methods. An example of algorithm development using weakest preconditions is presented, with comments on the role an interactive, computerized development system could play in this process. Finally, an overview is given of some of the problems encountered in automating such a methodology.

Parallelle Logika Programmering

A E G Potgieter
University van Pretoria

Abstract

Daar is verskeie benaderings tot die parallelle verwerking van logikaprogramme waarvan EN-parallelisme, STROOM-parallelisme en OF-parallelisme die belangrikste is. Tydens EN-parallelisme kan datakonflikte ontstaan, en gesofistikeerde kontrole word benodig om hierdie konflikte te verhoed.

Huidige navorsing in die gebruik en implementasie van parallelle logika tale sal bespreek word met spesiale verwysing na die implementasie van EN-parallelisme in J S Conery se EN/OF prosesmodel, en die implementasie van STROOM-parallelisme in PARLOG, Concurrent Prolog en GHC.

'n Stelsel wat ontwikkel is op 'n netwerk van transputers, waar STROOM-parallelisme verkry word deur pyplyning, sal verder bespreek word.

Formalisation of the Entity-Relationship Model

H H Rennhackkamp
University of Stellenbosch

Abstract

The entity-relationship data model was originally presented by Chen; and many extensions and improvements have since been proposed. It is an improvement on the relational data model in that data is viewed in the more natural manner as consisting of entities and the relationships between them, each with their descriptive attributes.

The relational data model has been formalised in many ways; from the original formalisation by its presenter Codd, in terms of data objects, operations and integrity rules. This formalisation has since been improved a number of times; also by Jacobs in terms of an extension of first order logic.

A similar extension of first order logic is presented as a formalisation of the entity-relationship data model.

Computer Animation and Applications

C F Scheepers
CSIR

Abstract

Developments in low-cost high-end graphics workstations have placed the many applications of computer animation within our reach. A combined use of these workstations with current video technology provides enough "hardware" capabilities to do animations. Animation software systems developed overseas are also available, thus only a lack of funds and possibly of imagination now prevent those interested to create their own computer generated animation sequences.

This paper will discuss computer animation and associated problems. Various applications of computer animation are also mentioned, specifically in education, entertainment, advertising and visualization in science. The discussion will be highlighted with a computer animation presentation.

Intelligent Computer Aided Instruction

T Thomas
University of South Africa

Abstract

The aim of intelligent computer aided instruction is to create systems that can teach and adapt to students on a one to one basis as a human teacher can in the same situation. Each teaching step is not hardcoded into the system but is decided upon based on the student's model at a particular time and the work that has been done previously. This paper discusses intelligent computer aided instruction in general and refers to a proposed system the author hopes to build to implement some of the ideas of intelligent tutoring. The first part of the paper discusses what intelligent CAI is and how it differs from conventional CAI. A discussion of the use of the overlay method for student modelling follows with reference to the work that has been done and the work proposed. The use of artificial intelligence techniques in the creation and use of teaching strategies based on the overlay model is then discussed with a description of some of the work planned in this area. The necessity of an authoring system that can create intelligent systems is then introduced.

Distributed Database Recovery

H Viktor
University of Stellenbosch

Abstract

The NRDNIX-system is a distributed database management system prototype currently being developed at the University of Stellenbosch. One of the main advantages of a distributed database over centralized counterpart is that nonfailing sites continue operating in the event of other site failure(s). This complicates the recovery process and it is therefore essential that robust recovery techniques must be implemented that ensure the global consistency of the database at all times. This is achieved by combining local recovery techniques and additional methods developed to facilitate distributed recovery.

In NRDNIX data is dynamically duplicated as needed. It is essential that these multiple copies have to be kept consistent even in the event of failure of one or more sites. This further complicates the recovery process and emphasises the need for the provision of a globally consistent view of the database at all times.

An overview of the factors which have to be brought into consideration when developing a

globally consistent system is given, as well as the recovery techniques implemented to provide a globally consistent distributed database.

Research on Intelligent Computer-Assisted Instruction (ICAI)

J van den Berg
University of Pretoria

Abstract

ICAI research lies at the intersection of the domains of artificial intelligence, cognitive psychology and computer aided instruction. The aim of ICAI is to develop adaptive instructional systems based on a diagnosis of the student's learning needs. Current ICAI systems use AI programming techniques and are usually implemented in languages such as LISP and Prolog. These systems usually consist of

- a problem-solving or expertise module
- a student model
- a tutorial module
- a communication module

Due to the size and complexity of ICAI systems, most current systems only focus on one or two of the above mentioned components.

The aims of this research are

- to critically evaluate the theoretical base, instructional principles and development process used in current ICAI systems
- to identify related areas of study from which known principles can be used in the development of ICAI systems
- to propose a methodology for designing and implementing ICAI systems as complete instructional systems
- to implement aspects of the research in a practical ICAI system
- to identify an area of ICAI (or a related topic) for PhD studies.

An Overview of Image Processing Environments

J Smit
CSIR

Abstract

The last two decades have seen image processing systems migrate from mainframe to workstation and micro to be used for an ever-increasing range of applications. This paper gives a brief overview of image processing applications and techniques. It traces the development of image processing environments, discusses some of their design issues and finally describes the functional components of an image processing environment.

A Frame- and Rule-Based Approach for Developing Medical Expert Systems

J van Jaarsveld
CSIR

Abstract

To date most local expert systems use the rule-based approach and few incorporate the frame approach. This paper describes the development of a medical expert system for interpreting the effects of hypertension using a combination of frames and rules. The structure and properties of the system are described by using the Arity Prolog/Expert Development System. In addition, the inference process linking the frames and the rules is discussed. Finally, some aspects of the knowledge engineering experience are detailed.

Determining Truth Values of Quantified Statements

A P Viljoen
University of South Africa

Abstract

In classical logic there exists a standard procedure to follow when one wants to determine the truth value of a given formula in a specific interpretation.

A new procedure, equivalent to the standard one, is proposed. This procedure also allows the calculation of truth values for formulas containing fuzzy predicates. A fuzzy predicate is interpreted as a fuzzy subset of a universe of discourse. Limiting the interpretation to a finite set, the quantifiers "for all" and "there exists" as well as nonstandard quantifiers like "few", "many" and "most" can be handled in a uniform manner.

The Application of Real-time Design Techniques to Simulation

G C Wells
Rhodes University

Abstract

This paper discusses the application of the Ward and Mellor design methodology for real-time systems to the field of simulation. The tools and heuristics used by Ward and Mellor are extended to provide a useful methodology for the design of real-time simulations. This is illustrated by the example of a real-time simulation of a manufacturing plant and process control system.

NOTES FOR CONTRIBUTORS

The purpose of the journal will be to publish original papers in any field of computing. These may include research, review and exploratory articles of interest to the journal's readers. The preferred language of the journal will be English, although papers in Afrikaans or other congress languages of IFIP will not be precluded. Typed manuscripts for review should be submitted in triplicate to:

Professor D G Kourie
Department of Computer Science
University of Pretoria
Hatfield 0083
Pretoria

Form of Manuscript

Manuscripts for review should be prepared according to the following guidelines.

- Use double-space typing on one side only of A4 paper, and provide wide margins.
- The first page should include:
  - title (as brief as possible)
  - author's initials and surname
  - author's affiliation and address
  - an abstract of less than 200 words
  - an appropriate keyword list
  - a list of Computing Review Categories.
- Tables should be typed on separate sheets of A4 paper, and should be numbered and titled.
- Figures should also be supplied on separate sheets of A4 paper, and should be identified on the back in pencil with the author's name and the figure number. Original line drawings, and not photocopies, should be submitted.
- Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin. Distinguish clearly between:
  - upper and lower case letters
  - the letter O and figure zero
  - the letter I and the number one
  - the letter K and kappa.

  References should be listed after the text in alphabetical order of the (first) author's surname, cited in the text in square brackets. References should take the following form:
  [1] E Ashcroft and Z Manna, [1972], The translation of 'GOTO' programs to 'WHILE' programs, *Proceedings of IFIP Congress 71*, North-Holland, Amsterdam, 250-255.
  [2] C Bohm and G Jacopini, [1966], Flow diagrams, Turing machines and languages with only two formation rules, *Comm. ACM*, 9 (3), 366-371.

Manuscripts accepted for publication should comply with the above guidelines, but may be in one of the following three formats:

- typewritten and suitable for scanning
- provided as an ASCII file on diskette
- camera-ready.

Authors wishing to provide camera-ready copy may obtain a page specification from the production editor.

Charges

A page charge, scaled to reflect production costs, will be levied on papers accepted for publication. The costs per final page are as follows:

Typed format, not camera-ready R60
Disk in ASCII format R40
Camera-ready format R20

These charges may be waived upon request of the author, and at the editor's discretion.

Proofs

Proofs of accepted papers will be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Copyright

Copyright in published papers will be vested in the publisher.

Letters and Communications

Letters to the editor will be welcomed and will provide a forum for discussion on topical issues. They should be signed, and should be limited to about 500 words.

Communications reflecting minor research contributions will be considered for publication in a separate section of the journal. Such communications will, however, not be regarded as a fully-fledged publication for FRD subsidy purposes.

Book Reviews

Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

Advertising

Placement of advertisements at R1000 per full page per issue and R500 per half page per issue will be considered. Enquiries should be directed to the production editor.